



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/676,658	09/30/2003	Zhenhai Lin	50277-2280	8991
42425 7590 02/07/2007 HICKMAN PALERMO TRUONG & BECKER/ORACLE 2055 GATEWAY PLACE SUITE 550 SAN JOSE, CA 95110-1089			EXAMINER VAUTROT, DENNIS L	
			ART UNIT 2167	PAPER NUMBER
SHORTENED STATUTORY PERIOD OF RESPONSE			MAIL DATE	DELIVERY MODE
3 MONTHS			02/07/2007	PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

<b>Office Action Summary</b>	<b>Application No.</b> 10/676,658	<b>Applicant(s)</b> LIN, ZHENHAI	
	<b>Examiner</b> Dennis L. Vautrot	<b>Art Unit</b> 2167	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

#### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

- 1) ☒ Responsive to communication(s) filed on 13 November 2006.
- 2a) ☒ This action is **FINAL**.                      2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

- 4) ☒ Claim(s) 1, 2, 4-10, 12-14, 16-22, 30, 31, 33-39, 41-43 and 45-51 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☒ Claim(s) 1, 2, 4, 5, 12 - 14, 16, 17, 30, 31, 33, 34, 41 - 43, 45, 46 is/are allowed.
- 6) ☒ Claim(s) 6 - 10, 18 - 22, 35 - 39, & 47 - 51 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \*    c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

- |  |   |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892)   | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                       | 5) <input type="checkbox"/> Notice of Informal Patent Application                       |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

## **DETAILED ACTION**

### ***Response to Amendment***

1. The applicants' amendment, filed 13 November 2006, has been received, entered into the record and considered.
2. As a result of the amendment, claims 1, 6, 10, 12, 15, 18, 22, 30, 31, 33-39, 41-43 and 45-51 are amended. Claims 3, 11, 15, 23-29, 32, 40, 44 and 52 have been canceled. Claims 1, 2, 4-10, 12-14, 16-22, 30, 31, 33-39, 41-43 and 45-51 are pending in the application.

### ***Response to Arguments***

3. Applicant's arguments, see page 20 – 21 of the response, filed 13 November 2006, with respect to claims 1 – 5 have been fully considered and are persuasive. The rejections of claims 1 – 5 have been withdrawn.
4. Applicant's arguments filed in regards to independent claims 6 and 18 have been fully considered but they are not persuasive.
5. Applicant's first and second arguments: "Menon does not disclose reading the catalog table as a response to a request" is not persuasive. The context of the reading in Menon is based on a call to the vault to read an asset. Examiner interprets the call to

Art Unit: 2167

the vault to read the asset to determine the attributes as being "in response to a request". Additionally, see column 17, lines 10 – 12 "...calls to the Vault API 106 may be made directly by an asset management tool..." This can be interpreted to be a request from the asset management tool. For the second argument that Menon does not disclose the formation of the data structure in memory as being initiated by a response to a request, it is not persuasive for the same reasons. In addition, see column 26, line 67 "AmsBase has methods for checking out the metadata of a data object from the Vault repository 108 into memory." It is well known in the art that process of "checking out" from a repository is made by a request.

6. Applicant's third argument: "Menon fails to disclose that the data structure inspected is not the same data structure created upon the initial reading of the catalog table." Examiner also finds this to not be persuasive. The concept of inspecting data structures without accessing the catalog table to determine the custom attributes of said particular object type is clearly shown in the language quoted. Taken in context with the rest of the reference, i.e. column 27, lines 16 – 19 "When a data object is checked out from the Vault repository 108 as an AmsBasePL object, the data objects metadata are manifested as a property list", this shows that the request referred to earlier (to check out), which precedes the construction in volatile in-memory data structures, occurs prior to the inspection of the data structures, as in the claim. For these reasons, Examiner is not persuaded by Applicant's arguments with regards to independent claims 6 and 18.

Art Unit: 2167

7. Applicant's arguments filed in regards to independent claims 10 and 22 have been fully considered but they are not persuasive. Specifically the argument that "No mention is made within the cited art of steps to be performed in *response to the application being launched*" is not persuasive. Right below the cited language from the rejection, another tool is described which Examiner interprets to be an application that is launched that causes the steps to occur. In column 17, lines 17 – 18 "An example of a read only tool is a movie player." When the movie player is launched the catalog is read, the volatile memory data structures are constructed and upon request to view the movie, the data structures are inspected without accessing the catalog table again. Additionally, the language cited in the first action on the merits shows that the steps are taking place by a storyboard and layout tool, which are also considered applications.

### ***Claim Rejections - 35 USC § 101***

8. The §101 rejections made in the previous office action, with the exception of the rejection for claim 31, which was not amended to include storage, have been withdrawn in light of the amendments.

### ***Allowable Subject Matter***

9. Claims 1, 2, 4, 5, 12 – 14, 16, 17, 30, 31, 33, 34, 41 – 43, 45, and 46 are allowed. The novelty in independent claim 1 relates to the upgrading portion of the claim. The combination of the following lines are what Examiner feels makes the claim unique: **"upgrading said application, wherein upgrading said application comprises the**

Art Unit: 2167

**steps of processing the data stored in the first table, wherein processing comprises: creating a first replacement table to hold the data from said first table; copying the data from said first table, wherein data from said one or more default attributes of said first object type is copied from said first table into said first replacement table; and deleting said first table; and retaining, in said third table, values for said first custom attribute of said first object type and said second custom attribute of said second object type.”** Based on the arguments made by Applicant, the prior art on record is considered distinguishable from the claims as amended. Copying data between tables in a repository and having a separate table for customized attributes are not considered novel, however, **the combination of the steps used to copy the tables for the purpose of upgrading an application was not found in the prior art.**

### ***Claim Rejections - 35 USC § 102***

10. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

11. Claims 6 – 10, 18 – 22, 35 – 39, and 47 - 51 are rejected under 35 U.S.C. 102(e) as being anticipated by **Menon** (US 6,615,204).

12. Regarding claims 6, 18, 35, and 47, **Menon** teaches a method for storing data in a repository comprising the steps of:

storing in a first table data for one or more default attributes of a first object type used by an application. (See fixed mapping asset table A FIG. 12, 1230);

storing in a second table, separate from said first table, data for one or more default attributes of a second object type used by said application. (See fixed mapping asset table B FIG. 12, 1231); and

storing in a third table, separate from said first and second tables, data for a first custom attribute of said first object type and data for a second custom attribute of said second object type, wherein said first custom attribute and said second custom attribute have the same data type (See FIG. 12, item 1106, where custom attributes are stored by the same type); and

storing, in a catalog table, data defining said first custom attribute of said first object type and said second custom attribute of said second object type (See FIG. 11, showing the custom attribute data being stored.)

wherein said catalog table stores data that identifies custom attributes for at least one object type (See FIG. 11, item 1105, note different types);

the method further comprises performing the following steps in response to a request to access an object instance of a particular object type:

Art Unit: 2167

reading said catalog table to determine the custom attributes of said particular object type (See column 17, lines 7-9 "For example, such tool 224 can be used to read or modify attribute values and/or to read an asset directly."); and

based on the information from said catalog table, constructing in volatile memory data structures that indicate the custom attributes of said particular object type (See column 27, lines 2-5 "Once in memory, a client uses accessor methods of AmsBase to get individual attributes of a data object. Contents may be checked out into memory or into memory or into a file in the local file system."); and

in response to a subsequent request to access an object instance of said particular object type, inspecting said data structures, without accessing said catalog table, to determine the custom attributes of said particular object type. (See column 27, lines 28-32 "Thus, an application program could at runtime, query this property list to determine the structure, i.e. attributes and types, and values of the object's metadata. Consequently, It is not necessary for the application to use built-in accessor methods to read the attributes.")

13. Regarding claims 7, 19, 36, and 48, **Menon** additionally teaches said catalog table includes: at least one first catalog column, wherein each row of said catalog table stores, within said at least one first catalog column, data that identifies an object type associated with the row (See column 20, lines 9 – 13 "Each box depicted within the second column 1105 represents a particular attribute for the associated asset [object].



Art Unit: 2167

Each attribute typically comprises three elements: (1) an attribute name; (2) an attribute value; and (3) an attribute type.” Here the type is part of the attribute field.),

at least one second catalog column, wherein each row of said catalog table stores, within said at least one second catalog column, data that identifies a custom attribute of the object type that is associated with the row (See column 20, lines 19-20 “Note that each entry can have a variable number of attributes within the second column.”), and

at least one third catalog column, wherein each row of said catalog table stores, within said at least one third catalog column, data identifying a data type of the custom attribute that is identified in said second catalog column (See column 20, lines 11 – 13 “Each attribute typically comprises three elements: (1) an attribute name; (2) an attribute value; and (3) an attribute type.” Here the type is also part of the attribute field.)

14. Regarding claims 8 and 37, **Menon** teaches the step of retrieving, in response to a request for an object instance of said first object type, the value of said first custom attribute associated with said object instance (See column 20, lines 9 – 13 “Each box depicted within the second column 1105 represents a particular attribute for the associated asset [object]. Each attribute typically comprises three elements: (1) an attribute name; (2) an attribute value; and (3) an attribute type.”); by performing the steps of: determining the data type of said first custom attribute from said third catalog column of a catalog-table row stored in said catalog table (See column 20, lines 9 – 13 “Each box depicted within the second column 1105 represents a particular attribute for

Art Unit: 2167

the associated asset [object]. Each attribute typically comprises three elements: (1) an attribute name; (2) an attribute value; and (3) an attribute type.” And lines 14-16 “The type is selected from a predefined list of types. As will be described below, each predefined type is associated with one of the separate metadata tables.” The type is also stored in the attribute field.); wherein: data in said at least one first catalog column of said catalog-table row matches data identifying said first object type (See FIG. 11, item 1114); and data in said at least one second catalog column of said catalog-table row matches data identifying said first custom attribute (See FIG. 11, item 1116); based on the data type of said first custom attribute, determining the identity of said third table (See column 20, lines 14-16 “As will be described below, each predefined type is associated with one of the separate metadata tales.”); and retrieving, from a value column of a third-table row stored in said third table, the value of said first custom attribute (See column 20, lines 11-13 “Each attribute typically comprises three elements: (1) an attribute name; (2) an attribute value; and (3) an attribute type.”), wherein: data uniquely identifying said object instance matches data in at least one instance-identifying column of said third-table row (See FIG. 11, item 1103); and data identifying said first custom attribute matches data in at least one attribute-identifying column of said third-table row (See FIG. 11, item 1103, where the object id matches).

15. Regarding claims 9, and 38, **Menon** additionally teaches the step of storing in said catalog table data defining a custom object type, separate from said first object type and said second object type, wherein the step of storing said custom object type

Art Unit: 2167

includes the step of inserting a row into said catalog table (See column 30, lines 43-45 “In this embodiment, the extensions (e.g., object B’ 1202b) are stored using asset table 1102 of the flexible mapped portion.”), wherein said row includes: within said at least one first catalog column, data that identifies said custom object type (See column 20, lines 11-13, specifically (1) name); within said at least one second catalog column, data that identifies a custom attribute of said custom object type (See column 20, lines 1 and 2, specifically object id), and within said at least one third catalog column, data identifying a data type of said custom attribute that is identified in said at least one second catalog column (See column 20, lines 11-13, specifically (3) type).

16. Regarding claims 10, 22, 39, and 51, **Menon** teaches a method for storing data in a repository comprising the steps of:

storing in a first table data for one or more default attributes of a first object type used by an application. (See fixed mapping asset table A FIG. 12, 1230);

storing in a second table, separate from said first table, data for one or more default attributes of a second object type used by said application. (See fixed mapping asset table B FIG. 12, 1231); and

storing in a third table, separate from said first and second tables, data for a first custom attribute of said first object type and data for a second custom attribute of said second object type, wherein said first custom attribute and said second custom attribute have the same data type (See FIG. 12, item 1106, where custom attributes are stored by the same type); and

storing, in a catalog table, data defining said first custom attribute of said first object type and said second custom attribute of said second object type (See FIG. 11, showing the custom attribute data being stored.)

wherein said catalog table stores data that identifies custom attributes for at least one object type (See FIG. 11, item 1105, note different types);

said catalog table stores data that identifies custom attributes for a plurality of object types (See FIG. 11, item 1105, note different types);

the method further comprises performing the following steps in response to said application being launched:

reading said catalog table to determine custom attributes from said plurality of object types (See column 17, lines 7-9 "For example, such tool 224 can be used to read or modify attribute values and/or to read an asset directly."); and

based on the information from said catalog table, constructing in volatile memory data structures that indicate the custom attributes of each of said plurality of object types (See column 27, lines 2-5 "Once in memory, a client uses accessor methods of AmsBase to get individual attributes of a data object. Contents may be checked out into memory or into memory or into a file in the local file system."); and

in response to a request to access an object instance of a particular object type of said plurality of object types, inspecting said data structures, without accessing said catalog table, to determine the custom attributes of said particular object type (See column 27, lines 28-32 "Thus, an application program could at runtime, query this property list to determine the structure, i.e. attributes and types, and values of the

Art Unit: 2167

object's metadata. Consequently, It is not necessary for the application to use built-in accessor methods to read the attributes.”)

17. Regarding claims 20 and 49 **Menon** teaches the step of retrieving, in response to a request for an object instance of said first object type, the value of said first custom attribute associated with said object instance (See column 20, lines 9 – 13 “Each box depicted within the second column 1105 represents a particular attribute for the associated asset [object]. Each attribute typically comprises three elements: (1) an attribute name; (2) an attribute value; and (3) an attribute type.”); by performing the steps of:

determining the data type of said first custom attribute from said third catalog column of a catalog-table row stored in said catalog table (See column 20, lines 9 – 13 “Each box depicted within the second column 1105 represents a particular attribute for the associated asset [object]. Each attribute typically comprises three elements: (1) an attribute name; (2) an attribute value; and (3) an attribute type.” And lines 14-16 “The type is selected from a predefined list of types. As will be described below, each predefined type is associated with one of the separate metadata tables.” The type is also stored in the attribute field.); wherein:

data in said at least one first catalog column of said catalog-table row matches data identifying said first object type (See FIG. 11, item 1114); and data in said at least one second catalog column of said catalog-table row matches data identifying said first custom attribute (See FIG. 11, item 1116);

based on the data type of said first custom attribute, determining the identity of said second table (See column 20, lines 14-16 "As will be described below, each predefined type is associated with one of the separate metadata tables." The second table here represents the custom attribute tables.); and retrieving, from a value column of a third-table row stored in said second table, the value of said first custom attribute (See column 20, lines 11-13 "Each attribute typically comprises three elements: (1) an attribute name; (2) an attribute value; and (3) an attribute type."), wherein:

data uniquely identifying said object instance matches data in at least one instance-identifying column of said second-table row (See FIG. 11, item 1103); and

data identifying said first custom attribute matches data in at least one attribute-identifying column of said second-table row (See FIG. 11, item 1103, where the object id matches).

18. Regarding claims 21 and 50, **Menon** additionally teaches the step of storing in said catalog table data defining a custom object type, separate from said object type, wherein the step of storing said custom object type includes the step of inserting a row into said catalog table (See column 30, lines 43-45 "In this embodiment, the extensions (e.g., object B' 1202b) are stored using asset table 1102 of the flexible mapped portion."), wherein said row includes: within said at least one first catalog column, data that identifies said custom object type (See column 20, lines 11-13, specifically (1) name); within said at least one second catalog column, data that identifies a custom attribute of said custom object type (See column 20, lines 1 and 2, specifically object id),

Art Unit: 2167

and within said at least one third catalog column, data identifying a data type of said custom attribute that is identified in said at least one second catalog column (See column 20, lines 11-13, specifically (3) type).

### ***Conclusion***

**THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Art Unit: 2167

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dennis L. Vautrot whose telephone number is 571-272-2184. The examiner can normally be reached on Monday-Friday 9:00-6:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cottingham can be reached on 571-272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Dv  
26 January 2007



CHRISTIAN CHACE  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100

